DIGITAL DATA LOGGING SYSTEM

A thesis submitted
In partial fulfilment of the requirements
for the degree of

MASTER OF TECHNOLOGY IN ELECTRICAL ENGINEERING

by
VINOD KUMAR PANDEY

to the

Department of Electrical Engineering
Indian Institute of Technology Kanpur

August 1972

CERTIFICATE

CERTIFIED that this work on "DIGITAL DATA LOGGING SYSTEM" has been carried out under my supervision and that it has not been submitted elsewhere for a degree.

B Prasada

Dr(B. PRASADA)
Professor,
Electrical Engineering Dept.
Indian Institute of Technology
Kanpur-16

ACKNOWLEDGEMENTS

I am highly indebted and grateful to Professor B. Prasada for his invaluable guidance and encouragement throughout the project. I would also like to express my deep sense of gratitude to Professor H.N.Mahabala for his fruitful discussions and facilities provided during this project.

I would like to extend my thanks to IBM Engineers, Mr. N.Saxena and Mr.R.D. Hinduja, for their help in learning the IBM 1800 system. My friends, Mr.R.K.Manocha, Mr. P.V.H.M.L. Narsimham and Mr. Ranjit Singh, deserve thanks for their helps from time to time.

Finally I would like to appreciate the sincere effort of Mr. R.Pandey for typing the manuscript.

VINOD KUMAR PANDEY

TO BINDA

ABSTRACT

A scheme of digital data logging for the analysis of fading phenomena on troposcatter links have been discussed. The data logging system uses IBM 1800 DATA AQUISITION AND CONTROL SYSTEM for ON-LINE aquisition of data. The computer time is shared between other Non-Process and Process programs with the help of TIME SHARING EXECUTIVE SYSTEM. The data is finally stored on magnetic tape which can be processed on IBM 7044/1401 systems.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

An experimental Troposcatter link is being set up between Nainital and Kanpur. In the initial phases of research program, it is intended to gather fading statistics on this channel at 2100 MHz. Such data is not yet available for this region and extracted parameters from the experimental data are expected to yield useful information for optimal system design of troposcatter communication links. With this in mind, the need was felt for a Digital data logging system which could store the raw data on digital tape, compatible with IBM 7044/1401 systems, available at this institute.

FADING

Weak persistent fields, due to scattering from lower layers of atmosphere, called Troposphere, are observed beyond horizon. The path loss depends upon the frequency and the distance between transmitter and receiver. The field strength at the receiver is acompanied by continuous fluctuations called FADES. Detailed experimental data on statistical distribution of fade durations and fading rates is sparse. Such data as is available, is reasonably consistent to the trend and order of magnitude but is not in good agreement with the shape of the distribution

Typically observed fading rates in Troposcatter channels range from 0.1 to 1.0 fades/second at 400 MHz and to 1 to 10 fades/second at 4000 MHz. In individual experiments however, the fade rates may vary from these values by a factor of 10 in either direction.

The entire fading spectrum is roughly subdevided into the following three groups.

(i) FAST FADES

(ii) SLOW FADES

(iii) VERY SLOW FADES

Fast fading, which is believed to be due to multiple path effects, generally has a Rayleigh distribution and its sampling time has been found to vary from 1 to 5 minutes. A slow fading which is believed to be due to changes of retractive index of the atmosphere is superimposed on fast fading.

The median of received signal over a long period of time, of the order of hours, is found to obey a log normal distribution i.e.; the values measured in db appear to obey a Gaussian distribution. Deviations from Rayleigh distribution in short term fading are occasionally observed because of other mechanisms of propogation like'ducting' and nonstationarity during the measured period.

Very slow fades, persisting over a long period of time of the order of a fraction of a day and longer, are also observed in troposcatter channels. These fades are characterized by the diurnal and seasonal variations of the long term median of the received signal.

STATISTICS OF FADING DATA

For system design of the data logging system it is necessary to have specifications of fading envelope in terms of its bandwidth and dynamic range. The upper limit of the bandwidth of fading is fixed by the maximum fading rate of interest. We fix our upper limit as 100 Hz. For slow fades only frequencies below 10 Hz are of interest to us.

Sampling time for short term statistics should be of the order of 1 to 5 minutes. For hourly variation, samples should be taken every hour and for diurnal and seasonal variations for a period of at least one year.

Besides the long-term median path loss over the channel, median signal strength over a short time is governed by the depth of fades. Since the seasonal variations have a dynamic range upto 40 to 50 db and the diurnal variations of hourly median are also about 20 db at the most, the system should therefore be able to handle a dynamic range of at least 70 db.

A SCHEME FOR DATA LOGGING

Our aim is to design a Data Logging System for the fading data as described above. Various systems have been proposed in past[1] and one of them was designed also. They used a strip chart recorder for on-line recording of data and ADC for digitizing it. But the IBM 1800/DACS, an on-line data aquisition and control system, recently acquired by IITK computer centre, can be very effectively used for this purpose. The IBM 1800/DACS has an Analog to Digital Converter (ADC) which can accept as many as 1024 analog input channels through a multiplexer. At this installation, only 16 analog input channels are available.

The Priority Interrupt Structure, which is a feature of IBM 1800/DACS, enables us to execute programs on a priority basis. The computer also supports a magnetic tape unit with 7-track read/write head,compatible with IBM 7044/1401 systems. Keeping these facilities in mind, we propose to use IBM 1800/DACS for our Data Logging System.

For IBM 1800/DACS, there are two programming systems available viz; Multiprogramming Executive (MPX) and Time-sharing Executive (TSX). MPX, though a avery powerful system for on-line applications, requires large core memory, at least 24 K word. The MPX system, generated at this installation with bare minimum facilities takes 11100 words of core, leaving only around 5000 words for users which is insufficient for a Data Logging System. Hence we have used TSX for our purpose. The executive in this system takes around 7000 words, leaving rest of 9000 words of core for users.

The data logging program consists of subprograms which can be modified to give different sampling rates and number of samples. The programs are also flexible and new subroutines can be added and/or deleted, if desired, to make the data logging programs very versatile.

Before describing the data logging system in detail, here are the specifications and restrictions for the system.

SPECIFICATIONS

(i) Fast fades have the maximum frequency components upto 100 cps. Thus the sampling rate for fast fades is 200 samples/second.

(ii) Fast fades are to be sampled for 1/2 minute. This gives 6000 samples.

(iii) Slow fades have the maximum frequency components upto 10 cps. Thus the sampling rate for slow fades is 20 samples/second.

(iv) Slow fades are to be sampled for 5 minutes. This gives 6000 samples.

(v) The dynamic range of signal amplitude is 70 db. The signal is passed through a logrithmic amplifier which converts the 70 db range into a linear range.

(vi) The data-logging routines should be executed at the highest priority available in IBM-1800.

(vii) The data is to be logged for 5.5 minutes every hour, throughout the year.

RESTRICTIONS

IBM 1800/DACS poses some restrictions on our data logging system.

(i) Due to small memory size and inadequate speed of I/O devices, not more than 6000 samples can be taken at a stretch.

(ii) The data has to be finally stored on to a magnetic tape. Since there is only one tape unit availabe, some operator interrention becomes inevitable for changing the magnetic tapes.

BLOCK DIAGRAM

The data-logging system was designed, keeping in view, all the specifications and restrictions as mentioned above. The overall picture of the system is shown in Fig.1. The output of the logarithmic amplifier is simultaneously fed to two active low pass filters with a 100 cps for fast fades and 10 cps for slow fades, as upper cutoff frequencies, respectively. The two types of fades are carried on two shielded pair cables to the IBM 1800 multiplexer terminals. Fast fades are connected to a solid state multiplexer point and slow fades to a Relay multiplexer point. The amplifiers shown in the figure provide sufficient gain so as to give an output of ± 5 volts dynamic range. The output stage amplifiers also act as buffers betweer the multiplexer terminals and the filters.

For the data logging routines to come into execution the IBM 1800 system is interrupted every hour. This interrupt is created by shorting a Process Interrupt Status Word (PISW) bit. To achieve this, we have used a battery driven electric clock, with a hole in the dial at 12th position. A photoresistance is attached at the back of the dial so that the light falls on it every minute of the hour excluding the 60th, when the minute needle is at the 12th position. This causes the resistance to vary, causing a relay to trip and shorting the PISW. Thus, we have the interrupt, created on the hour, every hour.

This external interrupt is connected at the highest priority level 00. The subroutine, included in the executive to servi œ this interrupt inturn creates an interrupt at level 05. The data logging routines are out of core disk resident routines which are brought into the variable core (VCORE) for execution whenever a programmed interrupt occures at level 05. At this time, the users program which is currently in execution is saved on to a disk area. The slow fades are then read for 5 minutes and this data is temporarily stored in VCORE. When the sampling of slow fades is finished, the data is transfered on to a disk file. The same procedure is followed for the fast fades. Every hour the data is transferred from disk to magnetic tape. Each set of 6000 data on the magnetic tape is labelled. The label contains the information about time, date, month and type of data i.e. slow or fast. For this purpose, the data logging system contains an inbuilt calander routine which keeps track of time, date and month. The provision for updating the calander in case of power failure etc. is also provided.

The whole system works automatically without any human intervention except for following occasions.

(i) The clock has to be synchronized by a similar clock at Nainital. This is done by synchronizing both the clocks independently with AIR time, everyday.

(ii) The appropriate tape has to be mounted every hour for transferring the data from disk to tape.

(iii) The calander is to be up dated during computer shut down.

CHAPTER 2

RELEVENT FEATURES OF IBM 1800 DACS

(1) Interrupt[2]

The interrupt feature in this system provides an automatic branch from the normal program sequence. It provides a means of switching from one program routine to another depending upon the external conditions. The Interrupt may be one of the following types

(1) Internal interrupt

These interrupts are created by any one of the following conditions:-

(a) An invalid operation code

(b) A parity error in transferring the data

(c) A storage protect violation

(d) A channel Address Register (CAR) check.

(2) Customer Engineer Interrupt

This interrupt is the lowest priority interrupt which is used by Customer Engineer for fault diagnostic purposes.

(3) External Interrupt

These interrupts are classifed in two major classes.

(a) Input/Output Interrupt

The I/O interrupts are created by the I/O devices attached to the system such as card read/punch, disk, magnetic tape etc.

(b) Process Interrupt[4,7]

The process interrupts are created by various processes external to the computer. Examples of this type of interrupt, are, rising temperature in a furnace or sudden increase in the pressure of a cylinder etc.

It is the process interrupt which is of very much interest to us because we are creating an external interrupt, every hour, to store the fading data.

Process Interrupts

There can be 24 levels of process interrupts in the system. This installation has 12 levels of interrupt. For each interrupt level there is a 16 bit Interrupt Level Status Word (ILSW). Attached to each ILSW bit , these can be either one Process Interrupt Status Word (PISW) or one Device Status Word (DSW). DSW is a status word associated with each I/O device. PISW is a 16 bit word where each bit can be connected to a process.

When a PISW bit is shorted by an external process, the corresponding ILSW bit is turned ON. In the IBM 1800 processor controller, there is a polling circuit which, after the end of the execution of each instruction, checks for an ILSW bit which is ON. When an ILSW bit is ON a forced Branch and Save Instruction Counter Indirectly (BSI Indirect) instruction is executed by the system. Every interrupt level braches through unique location in the core. This location contains the address of the LEVEL WORK AREA for that particular interrupt level.

In the level work area, the Instruction Register (I-Reg.) Accumulator, Accumulator extension (Q-Reg.), Index Register 1, (XR1) Index Register 2 (XR2), Index Register 3 (XR3) and the other indicators are saved. The control is then passed to Master Interrupt Control (MIC).

MIC senses the ILSW and branches to different routines depending upon which bit is ON, through Interrupt Branch Table (IBT).

For process interrupt an entry is made to Process Interrupt Entry point (PRIE). Here the PISW, corresponding to that level is sensed. The routine to service this PISW bit is then given the control. This is done with the help of Interrupt Core Load Table (ICLT).

The interrupt can be serviced by a subroutine in-cluded in the executive, a subroutine included with main line core load or a core load stored on to the disk by * STORECI Control Card. The ICLT entries for each FISW bit contain the information about the location of the interrupt servicing routine i.e. In-Executive, In-mainline or out of core. If the interrupt servicing routine is a out of core-coreload, the VCORE is saved in interrupt save area and the core load is brought in VCORE for execution. After the interrupt has been serviced the control is given back to MIC which resets the ILSW bit and reloads the previous program for execution. The flow of control during the service of an interrupt is shown in Fig. 2.

(b) ANALOG INPUT[2,3]

IBM 1800 DACS is capable of accepting Analog inputs. A physical phenomenon is first sensed and converted to an analog electrical signal by sensors or tranceducers. Low level signals must be amplified to a level acceptable for conversion to digital form. All lines from tranceducers are terminated at the control system on screw down terminals.

Conversion of Analog signals from a voltage level to digital information is accomplished by an Analog to-Digital Converter (ADC). Such converters are complex eneough so that if multiple sources of analog signals are to be converted, they share the use of one ADC.

The switching is accomplished by a multiplexer. There are two types of multiplexers available.

(1) Relay Multiplexer

The Relay Multiplexers accept an analog signal which is differential in nature and is within ± 5 volts dynamic range. The switching speed from one multiplexer point to another is 100 cps maximum. The same multiplexer point can be addressed with the maximum rate of 50 cps.

(2) Solid State Multiplexer.

These multiplexers can be used for high speed data aguisition systems, The same multiplexer point can be addressed with a maximum rate of 10,000 cps. The signal attached to these terminals should have one terminal grounded.

There can be as many as 1024 analog input terminals attached to one ADC. Out of these, only 256 terminals can be solid state. Each multiplexer point can be selected by an XlO instruction.

Analog-to-Digital Converter

The ADC provides the IBM 1800 with the ability to convert bipolar analog signals with in ±5 volt range to digital values. It includes a buffer amplifier and has program selectable resolutions of 8,11 and 14 bits. The ADC conversion time depends only upon the number of bits of output that are to be developed. Conversion times are as follows:-

Resolution	Conversion Time
8 bits	29 usec.
11 bits	36 usec.
14 bits	44 usec.

The input impedance of ADC is of the order of 10 Megaohms.

DATA word of an ADC

The data word developed in the ADC Register is compatible with IBM 1800 word format as shown in Fig. 3. The data word allows for sign plus 14 bit resolution. Conversion by stored program, of the output of the ADC should assume a position of the binary point. This position of the binary point does not change with the resolution of data. Only the number of significant bits in the ADC converted value changes. Negative numbers are in 2's compliment form

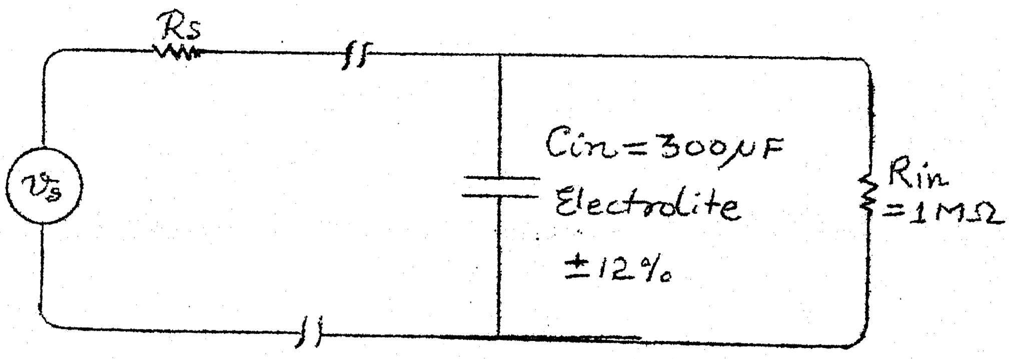
CHAPTER 3

THE DATA LOGGING SYSTEM

(1) SIGNAL CONDITIONING ELEMENTS.

The analog signals connected to the multiplexer terminals should have proper frequency response, source impedance and dynamic range. This is achieved by the two active low pass filters and the four amplifiers, Amp 1, Amp 2, Amp 3, Amp 4. This arrangement is shown in Fig.1. The input characteristics of the multiplexers pose some restrictions on the design of these amplifiers. We will discuss these restrictions one by one.

(i) Input to the relay multiplexer, as seen from the source terminals looks as follows:



where v_s and R_s are the input voltage and source resistance respectively.

The input resistance of the multiplexer is very large as compared to R_s and $\frac{1}{wC_{in}}$ at frequencies below 10 cps and is of the order of 1M ohms. Thus R_s and C_{in} combination works as a low pass filter, the upper cutoff frequency of which should be 10 cps. Hence the value of R_s can be calculated. It comes out to be 300 ohms.

But due to stray capacitance, cable resistance, leakage resistance of the capacitor, it has been experimentally found to be less than 50 ohms. So the output impedance of the Amp 4 should be less than 50 ohms.

(ii) The input voltage to the multiplexer should have a dynamic range of ± 5 volts, for maximum accuracy.

(iii) The output of the Amp 4, i.e. the slow fade signals should be differential in nature.

(iv) The output impedance of the Amp 3 should be between 100 to 1000 ohms.

The two low-pass filters should have the following characteristics.

(i) Filter 1 should have an upper cutoff frequency of 100 cps.

(ii) Filter 2 should have an upper cutoff frequency of 10 cps.

(iii) The signal strength above the cutoff frequencies for both the filters should fall down at the rate of at least 20 db/octave.

LOW PASS FILTERS[9]

The circuit diagram of the low-pass filters is shown in Fig. 5(a). The response is shown in Fig 5(b). The transfer function of the filter is

$$\frac{v_2}{v_1}(s) = \frac{1 + k\ T^2\ s^2}{s^2\ T^2 + 4Ts\ (1-m) + 1}$$

where

$$k = \frac{R_1}{R_1 + R_2} = .01$$

$$m = \frac{R_4 + R_3}{R_3} = .8$$

$T = RC =$ .10 for slow fades

.01 for fast fades.

Filter 1

C = .13 uF

R = 15 K

Filter 2

C=1.36 uF

R== 15 K

The cutoff slope obtained by one section of the filter is 14 db/octave.

We have used two sections of the filter to obtain the desired response of 28 db/octave.

INTERRUPT GENERATOR[8]

The interrupt in IBM 1800 is created by shorting a PISW bit. The circuit to achieve this is shown in Fig. 6. -------. When light is falling on the photoresistor, its resistance is approximately 500 ohms and since transistor T1 is in cutoff state, a very small current flows into the coils of Relay 1. The output terminals of Relay 1 remain open. The output transistor of monostable (T_3) is in saturation, forcing the transistor T_4 to remain in cutoff and thus Relay 2 remains open.

When the minute needle of the clock reaches the 12th position, the light falling on the photo resistor is blocked and its resistance increases. The minimum resistance measured experimentally is 20K. It goes as high as 50 K when shielded from stray light also. At this moment T_1 goes to saturation, Relay 1 trips and a sudden voltage, reduced in magnitude by the resistances R_1 and R_2, is applied to the monostable input which triggers the monostable which inturn forces T_3 to cutoff. The pulse width of monostable is 4 msec. For this time, T_4 goes to saturation and 4 mA of current flows into the coil of Relay 2. The Relay 2 trips. This shorts the PISW bit causing an interrupt in the IBM 1800 system.

After 4 msec. the transistor T_3 goes back to saturation, transistor T_4 goes to cutoff and Relay 2 terminals become open.

SOFTWARE

As discussed in the previous chapter, the clock and photocell arrangement generates an interrupt every hour. This interrupt is wired to FISW bit 00 at ILSW bit 00 The routines to perform the functions such as reading analog inputs, transfering data from memory to disk or disk to tape, can not operate at this interrupt level because all the I/O devices are attached to a lower interrupt level and they can not be used by a program operating at a higher level. The highest interrupt level which can utilize all the devices required to perform the operation of data logging is level 05. At the same time the data logging operation should be given priority over other programs being executed in VCORE. In order to resolve these conflicting requirements, a small subroutine has been included in the executive at the system generation time, to service an external interrupt at level 00 bit 00. The name of the routine is INEXC (IN-Executive). This routine creates a programmed interrupt at level 05.

When INEXC has been executed the interrupt level 00 is reset and the interrupt level 05 is serviced. The core load to service this interrupt is an 'Interrupt-coreload' stored on the disk. This core load is given a name 'SCATR'.

Before SCATR comes into VCORE for execution VCORE is saved on disk in INTERRUPT SAVE AREA, reserved on disk drive zero at the time of system generation. This area consistes of 9384 words which is exactly equal to the size of VCORE.

The core load SCATR consists of a main program and four subroutines.

(i) DIGIT (Main program)

(ii) SLOW

(iii) TRAN 1

(iv) FAST

(v) TRAN 2

DIGIT

The main program DIGIT controls the sequence of operations to be performed, as shown in Fig. 8. The sequence is as follows:

(i) To call subroutine SLOW to read slow fade data

(ii) To call subroutine TRAN 1 to transfer the slow fade data on to the disk.

(iii) To call subroutine FAST to read fast fade data.

(iv) To call subroutine TRAN 2 to transfer the fast fade data on to the disk.

(v) To queue the core load MATRN for manual mode data. transfer from disk to tape, if the fourth word in file HEADR is one.

(vi) To queue the core load AUTO 1 for automatic mode data transfer from disk to tape, if the fourth word in the file HEADR is two

(vii) To exit through MIC where level 05 is reset.

SLOW

The subroutine SLOW reads the relay multiplexer point 00 at every 50 msec. interval, for 6000 times. Data read is stored in the memory in a COMMON area called DATA. After reading the slow fade data for 6000 times the routine returns the control to main program DIGIT.

TRAN 1

The subroutine TRAN 1 transfers the data in COMMON area on to a permanent, file protected , disk file called SLOW. The disk file is created by*STORE DATA control card under the control of DUP. The file SLOW consists of 20 records of 300 word each. After transfering the slow fade data on to the disk file the subroutine transfers the control back to main program DIGIT.

FAST

This subroutine reads the solid-state multiplexer point /13E8 for fast fade data, at 05 msec interval for 6000 times. The data read is stored in the same common area DATA. After reading 6000 samples, the subroutine transfers conrol back to main program DIGIT.

TRAN 2

This subroutine transfers the fast fade data in VCORE on to a permanent, file protected disk file,called FAST. The disk file FAST consists of 20 seconds of 300 words each and is created by *STORE DATA control card. After the data has been transferred from memory to disk, the subroutine returns the control back to main program.

MATRN

The program is queued on basic level with priority1. Since only single priority Queues are allowed in the present system the programs in queue are served on first come first served basis . When digit returns control to MIC, MIC checks for any recorded interrupts. Since there was no interrupt waiting for service, the control is given to program sequence Control,(PSC). PSC checks the queue for any program waiting there. Then MATRN comes into execution.

MATRN is a main line core load which transfer the data from the disk files onto a magnetic tape. It uses three disk files.

(i) HEADR - For information about time,date and month.

(ii) SLOW - For slow fade data

(iii) FAST - For fast fade data.

As soon as MATRN comes into execution it types the following message on the console typewriter and comes to a **WAIT** state.

MOUNT THE TAPE FOR SCATTER COM. AND MAKE IT READY.

The operator has to mount the proper tape, make the tape ready and press start for further execution. This operator intervention is needed because some other user might be using the magnetic tape when the interrupt occures or the right tape used for data logging might not be mounted.

Note:- If the users program has processed n records and interrupt occurs, it is not possible for the DATA Logging routines to maintain the status of his tape at n+1th record after the data logging has been finished. It is the users responsibility to put his tape back, such that he can process the n+1th record.

MATRN calls a subroutine POSTN. This subroutine reads each header record and checks for a word having '$$'. **The** - $$ sign is an indication that the data has been written upto this point and the blank tape starts from here. The tape is back-spaced one record and control is given to main program MATRF.

MATRF first writes a header record for each 6000 data words. The header record consists of four words.

Word	Contents
1 -	Time in hours
2 -	Date
3 -	Month
4 -	1 for slow fade data
	2 for fast fade data

After writing the header record the data is read from disk files SLOW & FAST, and written on the tape. There are 20 data records of 300 words each for each type of fade, per hour.

At the end of data transfer a trailer record is written. This record consists of 4 words. The first word contains $$ and the rest 3 words are redundant. They are added to make the word lenght of header and trailer records equal. After the completion of this operation the following message is printed.

REMOVE THE TAPE AND SAVE IT

and control is transferred back to PSC.

CALENDER

The Data logging system has an in-built calender which keeps track of Time,Date and Month. The time is increamented by one, every hour, with the help of a COUNT-routine. The COUNT routine is a subroutine which can call itself after a specified period of time. The general calling sequence of the count routine is as follows:

CALL COUNT (II, IN, INTVL)[4]

where II is the number specified for the count routine. This number is specified at system generation time, by the *INCLD control card. IN is the programmed timer which is in use for this count routine. There are in all 9 programed timers available. Each timer has a time base of 1 second. INTVL is an integer which is decreased by one, at every timer interrupt. When the counter INTVL becomes zero, an interrupt is created at timer level, i.e. at level 0, and the count routine included in the executive is executed. In our case

IT	=	1
IN	=	4
INTVL	=	3600

The count routine included in executive, creates an interrupt at level 10, every hour. The CLNDR core load services this interrupt and increases the Time by one. It also increases the date and month accordingly.

TO START THE TIMER

Before the COUNT routine resumes its repeatetive operation, it has to be started. This is done by a core load SCNT. The core load SCNT comes in execution whenever console interrupt is pressed with SENSE SWITCH 7 OFF and DATA SWITCH 0 ON.

CLNDR

It is an interrupt core load. It updates the disk file HEADR which stores the information about time,date and month. HEADR is a permanent, file protected disk file. It consists of one record of 4 words.

Word	Contents
1	Time in hours
2	Date
3	Month
4	1= Manual-Mode, 2 = Auto-Mode

The core load can operate un interrupted for 3 years continuously. February is always considered to be of 28 days only. It increments the hour by 1 and accordingly dates and months are incremented at the appropriate time.

SAMPLING RATE

The slow fade and fast fade date are to be sampled at 50 msec. and 5 msec intervals each. This sampling time is decided by the routines SLOW and FAST themselves. The technique used to achieve this is discussed here.

While reading an analog input, the flow chart shown in Fig. 7 has been used.

The time elapsed between two samples is equal to the time taken by the program to execute the instructions from A to B and back to A. T1 is the time for fast fades to go from A to B. It is 14.7 m secs. T2 is the time for slow fades to go from A to B. It is 2.3 msec . To make T1=50 m.sec. and T2=5 m.sec., some redundant instructions have been introduced in the return path. The instructions are

```
LØØF      MDX      2      -1
          BSC             LØØP
```

These two instructions take 10 usec. for execution. The index Register 2 is loaded with appropriate value, so that these two instructions are executed for 36.3 and 1.7 msec. for slow and fast fades respectively

CORE LOAD INTO7

This core load comes into execution, whenever console interrupt is pressed with sense switch7 OFF. It checks the DATA switches 0,1,2,3 and 4, and queues different core loads, depending upon which data switch is ON. The various options that are available are as follows:

DATA SWITCH ON	OPERATION
0	START COUNT ROUTINE (core load SCNT)
1	START AUTOMATIC MODE (core load AUTOS)

2	START MANUAL MODE (core load MMODE)
3	Update Calender (core load VINOD)
4	START CONTINUOUS SCANNING (core load CONT)

The flow diagram is shown in Fig. 10.

CORE LOAD AUTOS

The core load reads the file HEADR from the disk and makes its fourth word =2 . When the program DIGIT tests this word and finds it equal to 2, it queues the core load AUTO1 for transfering the data on tape, without operator intervention.

CORE LOAD MMODE

It reads the file HEADR from the disk and makes its fourth word equal to one. When program DIGIT tests for this word and finds it equal to one, the data transfer from disk to tape requires operator intervention.

CORE LOAD CONT

This core load operates in following sequences of operations.

1. Subroutine SLOW is called to read the slow fade data.

2. Subroutine TRAN1 is called to transfer slow fade data on to the disk.
3. Subroutine FAST is called to read fast fade data.
4. Subroutine TRAN2 is called to transfer the fast fade data on to disk.
5. Core load AUTO1 is quened to transfer the data from disk to tape without operator intervention.
6. Core load CONT itself is queued for repeating operations 1 to 5.

UPDATING THE CALENDER (core load VINOD)

Since computer might be shut down due to various reasons like power failure, airconditioner failure, or some hardware failure in the system, the data logging system during that period will not be operating and hence the file HEADR, which contains the Time, Date and Month will not be updated. To update the file HEADR as desired by the operator, a program named VINOD has been stored permanently on to the disk. The core load VINOD reads the input from the type-writer. The operator has to enter the hour, of the day, month and date through the key board, and the core load will update the file HEADR.

COLD START (program COLD1)

The program is used as a cold start program. It is the first program which is executed when the TSX system is loaded from disk to memory. The program unmasks all the interrupt levels and writes, the following message on the typewriter

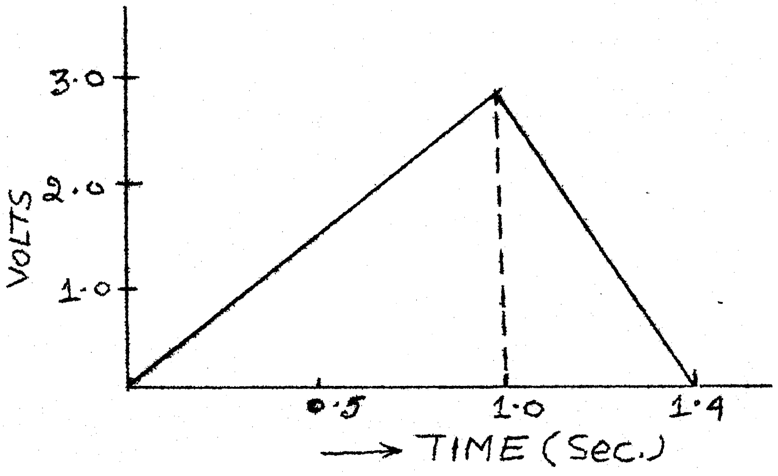
COLD START TEST

This program can be changed if the user decides to do so.

CHAPTER 4

PERFORMANCE OF SYSTEM AND CONCLUSION

The whole data logging system was tested in a simulated ON-line environment. A user's program was being compiled when the interrupt was created. A triangular waveform was connected to the relay multiplexer terminals. The wave-form was generated by HP low frequency signal generator. The shape of the waveform is shown below.



The data was stored on to a magnetic tape and was printed on the type-writer. The subroutine to perform this function is listed in Appendix C. The samples showed a periodicity. There were 27 samples per period which shows that the sampling rate and the oscilloscope time measurement match with in 3%. The sample values for a typical period are shown in Table 1 . The calculated rize in sample voltage is .185 volts. In the linear region of the waveform, the rizes are within ± 4% of this value. Which shows that samples are taken at equal time intervals.

A similar experiment was carried out for fast fades. The triangular waveform has a time period equal to .15 seconds. The periodicity of the samples showed that these were exactly 30 samples per period. This is in good agreement with the theoretical value. The sampling time was found to be uniform within ± 4%.

When the data logging was finished the user's program resumed its execution.

APPENDIX A

OPERATING PROCEDURES

The DATA LOGGING SYSTEM requires some operator intervention for operating effectively. We have discussed here, different steps which are to be taken by the operator, depending upon the different external conditions.

(1) External Connections

(i) The output of the interrupt generating circuit as shown in Fig. 1 should be connected to PISW bit 00.

(ii) The output of the Amp 3 as shown in Fig. 1 is connected to Relay Multiplexer point 00.

(iii) The output of Amp 4 as shown in Fig. 1, is connected to Solid State Multiplexer point. This point is located in 1826 enclosure, just below the Analog voltage calibrator.

(2) Cold Start

When the IBM 1800 system is shut down, certain procedure has to be followed to bring the TSX system from the disk into the core memory. This procedure is called cold start. The steps to be followed are as follows.

(i) Put the system power ON.

(ii) Mount the TSX ON-LINE SYSTEM disk cartridge on the disk drive zero and make it ready.

(iii) Clear the core memory to zeros.

(iv) Put the three TSX cold start cards into the card reader hopper and make the card reader ready

(v) Press PROGRAM LOAD switch on the console.

Following message is printed on the system typewriter

```
COLD START TEST
```

All this time the computer comes to wait state

(3) To Start the Data Logging Clock

After the cold start, when the system is in wait state, put the SENSE SWITCH 7 OFF and DATA SWITCH 0 ON and press Console Interrupt. Following message is printed.

```
COUNT ROUTINE NO. 1 HAS BEEN STARTED
 nn      mm      11      k
```

where nn is the time; mm is the date, 11 is the month and k is equal to 1.

The count routine no. 1 has to be started on following circumstances.

(1) ON system RELOAD

(2) ON system RESTART

(3) ON COLD START

(4) Manual Mode Operation

When ever the tape unit is being used or it is expected to be used when the interrupt comes and the operator is present on the system, the DATA LOGGING system should be operated in manual mode.

To put the Data logging system into manual mode, following procedure should be followed

(1) Put the SENSE SWITCH 7 OFF

(2) Put the DATA SWITCH 2 ON

(3) Press CONSOLE INTERRUPT

following message is printed

```
TIME   DATE    MONTH   MANUAL MODE
 nn     ll      mm         k=1
```

where nn, ll, mm are time, date and month correspondingly.

When the Data logging system is in manual mode, the tape has to be mounted every hour or whenever the interrupt comes. Before the data is transferred on to the tape, following message is printed.

```
MOUNT THE TAPE FOR SCATTER COM. AND MAKE IT READY
```

and the computer comes to WAIT state

The operator has to mount the proper tape and ready the tape unit and then press the START SWITCH on the console.

When the data transfer is finished the following message is printed.

```
REMOVE THE TAPE AND SAVE IT.
```

At this time the operator has to remove the tape and keep it at a proper place.

(5) Automatic Mode Operation

When the Data logging system is in AUTO MODE it does not require any operator intervention. To put the system in automatic mode, following procedure has to be followed.

(i) Put the SENSE SWITCH 7 OFF

(ii) Put the DATA SWITCH 1 ON

(iii) Press CONSOLE INTERRUPT

following message is printed.

```
TIME DATE MONTH AUTO MODE
 nn   11   mm     2
MOUNT THE TAPE FOR SCATTER COM. AND MAKE IT READY
```

and the computer comes to a wait state. The operator has to mount the proper tape for data logging ready the tape unit and press START on the console. After doing this the data transfer on to the tape will be automatic.

But during the period when Data logging system is in AUTO MODE, the tape unit should remain ready and the tape should not be disturbed.

(6) Continuous Mode of Operation

To put the Data logging system in continuous mode, following procedure has to be followed.

(1) Mount a New tape on the tape unit and make it ready.

(2) Put SENSE SWITCH 7 OFF

(3) Put DATA SWITCH 4 ON

(4) Press CONSOLE INTERRUPT

Following message is printed after each 6 to 7 minutes.

```
DATA   TRANSFERRED   ON   TAPE

TIME   DATE   MONTH

 nn     mm      11
```

To stop the continuous mode of data logging press STOP RESET and START buttons on console, in the same sequence. The system is RELOADED and data logging stopes.

Note- In continuous mode of operation the Interrupt should be disconnected , because it does not require an Interrupt.

(7) Updating the Calender

For updating the time, date and month, put the sense switch 7 OFF, data switch 3 ON and press console Interrupt. The data is entered through typewriter. The operator has to follow the instructions as being typed.

An illustrative example is shown in Appendix-B .

This procedure is to be followed whenever the computer is shut down for more than an hour.

(8) Processing a New Tape

A data record having '$$' as its first data word has to be written on every newtape, used for data logging. The procedure to do it is as follows.

(i) Mount the new tape and make it ready

(ii) Put the following two cards in card reader hopper and make it ready.

```
// b JOB            16
// b XEQ  b TEST    FX
```

The core load test is executed and the tape can be used now for data logging.

(9) The electric clock has to be updated every day by synchronizing it with AIR timings.

APPENDIX-B

```
Enter time in hours only
Example ** if it is 11 hours 43 minutes enter 11
09
Time entered is    9
If correct type yes if not type   not
Yes
Enter date
13
Date entered is    13
If currect type  yes if not type not
Yes
Enter month in two digit number
Example ** July =07
08
Month entered is    8
If correct type  yes  if not type  not
Not
Enter month in two digit number
Example ** July=07
08
Month entered is  8
If correct type   yes if not type not
Yes
```

APPTENDIX C

(1) PROGRAM FOR INITIALIZING THE TAPE

```
// JOB
// FOR TEST
*IOCS(MAGNETIC TAPE)
*NON PROCESS PROGRAM
*ONE WORD INTEGERS
      DATAIDOLR/'$$'/
      II=0
      12=1
      13=3
      REWIND 14
      WRITE (14)IDOLR,I1,12,13
      CALL EXIT
      END
// DUP
*STORECI                  TEST  TEST  TEST
*CCEND
```

(2)   THE CORE LOAD TO SERVICE CONSOLE INTERRUPT

```
// JOB
// FOR INTO7
*IOCS(TYPEWRITER,DISK)
        EXTERNALSCNT,AUTO,MANL,VINOD,CONT
99      FORMAT('ALL THE DATA ENTRY SWITCH BETWEEN OAND 4
       1 ARE OFF')
        CALL DATSW(O,I1)
        GOTO(1,2),I1
1       CALL QUEUE(SCNT,1,O)
        GOTO 100
2       CALL DATSW(1,I2)
        GOGO(3,4),I2
3       CALL QUEUE (AUTO,1,O)
        GOTO100
4       CALL DATSW(2,I3)
        GOTO(5,6),I3
5       CALL QUEUE (MANL,1,O)
        GOTO100
6       CALL DATSW(3,I4)
        GOTO(7,8),I4
7       CALL QUEUE(VINOD,1,O)
        GOTO100
8       CALL DATSW(4,I5)
        GOTO(9,I0),I5
```

```
9         CALL QUEU(CONT,1,0)
          GOTO100
10        WRITE(1,99)
100       CALL ENDTS
          CALL INTEX
          END
// DUP
*STORECI   I              INTO7  INTO7         2407
*CCEND
```

(3) CORE LOAD TO START COUNT ROUTINE

```
// JOB
// FOR SCNT
*LIST ALL
*IOCS(TYFEWRITER)
          CALL COUNT(1,4,[illegible])
          WRITE(1,100)
100       FORMAT('COUNT ROUTINE NO'1  HAS BEEN STARTED')
          CALL VIAQ
          END
// DUP
*STORECI   M           SCNT    SCNT    SCNT
*CCEND
```

(4) CORE LOAD FOR STARTING MANUAL MODE

```
// JOB
// FOR MMODE
*IOCS(TYPEWRITER,DISK)
      DEFINE FILE1802(1,4,U,IHEAD)
      IHEAD=1
      READ(1802'IHEAD)ITIME,IDATE,MONTH,INCTR
      INCTR=I
      IHEAD=1
      WRITE(1802'IHEAD)ITIME,IDATE,MONTH,INCTR
      WRITE(1,100)ITIME,IDATE,MONTH,INCTR
100   FORMAT('TIME DATE MONTH MANUAL MODE'/4I5)
      CALL VIAQ
      END
// DUP
*STORECI  M         MMODE MMODE MMODE
*FILES(1802,HEADR,0)
*CCEND
// DUP
*SEQCH MANL  MMODE,INTO7
```

```
(5)      CORE LOAD TO START AUTOMATIC MODE
// JOB
// FOR AUTOS
*IOCS(TYPEWRITER,DISK)
         DEFINE FILE1802(1,4,U,IHEAD)
         IHEAD=1
200      FORMAT('MOUNT THE TAPE FOR SCATTER COM AND READY IT')
         WRITE(1,200)
         PAUSE
         CALL POSN
         READ (1802'IHEAD) ITIME, IDATE,MONTH, INCTR
         INCTR=2
         IHEAD=1
         WRITE(1802'IHEAD)ITIME,IDATE,MONTH,INCTR
         WRITE(1,100)ITIME,IDATE,MONTH,INCTR
100      FORMAT('TIME DATE MONTH  AUTO MODE'/4I5)
         CALL VIAQ
         END
// FOR
         SUBROUTINE POSN
         DATA IDOLR/'$$'/
         DIMENSION IBUFR(4)
         REWIND I4
9        READ (I4)IBUFR
         IF(IBUFR(1)-IDOLR)10,12,10
```

```
10      DO 11 I=1,20
        READ(14)
11      CONTINUE
        GOTO9
12      RETURN
        END
// DUP
*STORECI  M        AUTOS AUTOS AUTOS
*FILES(1802,HEADR,0)
*CCEND
*SEQCH  AUTO  AUTOS,INT07
(6)     CORE LOAD FOR UPDATING CALANDER
// JOB
// FOR CLNDR
*IOCS(KEYBOARD)
*IOCS(TYPEWRITER)
*IOCS(DISK)
         DEFINE FILE1802(1,4,U,IHEAD)
         DATAND/'NO'/
100      FORMAT('ENTER TIME IN HOURS ONLY'/'EXAMPLE**IF IT IS
         ' 11 HOURS 43 MINUTES ENTER 12'/)
101      FORMAT(12)
102      FORMAT('TIME ENTERED IS' 4X,12/'IF CORRECT TYPE  YES
         1IF NOT TYPE  NOT'/)
```

```
103     FORMAT(A2,A1)
201     FORMAT('DATE ENTERED IS' 4X,I2,/'IF CRRECT TYPE YES
     1    IF NOT TYPE  NOT')
200     FORMAT('ENTER DATE')
300     FORMAT('ENTER MONTH IN TWO DIGIT NUMBER'/'EXAMPLE**
     1    JULY=07'/)
301     FORMAT('MONTH ENTERED IS'I2,/'IF CORRECT TYPE   YES
     1    IF NOT TYPE NOT'/)
        INCTR=1
11      WRITE(1,100)
        PAUSE
        READ(42,101)IBUF1
        WRITE(1,102)IBUF1
        PAUSE
        READ(42,103)IBUF2,IBUF3
        IF(IBUF2-NO)10,11,10
10      ITIME=IBUF1
21      WRITE(1,200)
        PAUSE
        READ(42,101)IBUF1
        WRITE(1,201)IBUF1
        PAUSE
        READ(42,103)IBUF2,IBUF3
        IF(IBUF2-NO)20,21,20
```

```
20         IDATE=IBUF1
41         WRITE(1,300)
           PAUSE
           READ(42,101)IBUF1
           WRITE(1,301)IBUF1
           PAUSE
           READ(42,103)IBUF2,IBUF3
           IF(IBUF2-NO)40,41,40
40         MONTH=IBUF1
           WRITE (1802'1)  ITIME,IDATE,MONTH,INCTR
           CALL VIAQ
           END
// DUP
*STORECI  M             VINOD CLNDR VINOD
*FILES(1802,HEADR,0)
*CCEND
```

(7)   CORE LOAD FOR CONTINUOUS SCANNING

```
// JOB
// FOR CONT
*IOCS(DISK)
           DEFINE FILE 1800(20,300,U.ISLOW)
           DEFINE FILE 1801(20,300,U.IFAST)
```

```
        EXTERNAL CONT
        EXTERNAL AUTO1
        COMMON/INSKEL/ISLOW,IFAST
        COMMON IDATA (6000)
        CALL SLOW
        CALL TRAN1
        CALL FAST
        CALL TRAN2
101     CALL QUEUE(AUTO1,1,0)
        CALL QUEUE(CONT,1,0)
        CALL VIAQ
        END
// DUP
*STORECI  M          CONT  CONT  CONT
*LOCALSLOW,TRAN1,FAST,TRAN2
*FILES(1800,SFADE,0)
*FILES(1801,FFADE,0)
*.CEND
```

(8)

```
// JOB
// FOR
        SUBROUTINE INEXC
        CALL ENDTS
```

```
      CALL LEVEL(5)
      CALL INTEX
      END
// DUP
*STORE              INEXC
```

(9)

```
// JOB
// * THE SUB ROUTINE IS ACOUNT ROUTINE FOR CALENDER
// FOR
      SUBROUTINE TIME
      CALL COUNT(1,4,3600)
      CALL LEVEL(I0)
      RETURN
      END
// DUP
*DELETE             TIME
*STORE              TIME
```

(10) THE CORE LOAD TO MAINTAIN CALENDER

```
// JOB
// FOR CLNDR
*IOCS(DISK,TYPEWRITER)
      DEFINE FILE1802(1,4,U,IHEAD)
```

```
4       FORMAT(4I5)
        IHEAD=1
        READ(1802'IHEAD)ITIME,IDATE,MONTH,INCTR
        IF(ITIME-24)2000,2001,2001
2000    ITIME=ITIME+1
        GOTO3000
2001    GOTO(1,2,1,3,1,3,1,1,3,1,3,1),MONTH
1       IF(IDATE-31)2002,2003,2003
2002    IDATE=IDATE+1
        ITIME=1
        GOTO3000
2       IF(IDATE-28)2002,2003,2003
3       IF(IDATE-30)2002,2003,2003
2003    IF(MONTH-12)2004,2005,2005
2004    MONTH=MONTH+1
        IDATE=1
        ITIME=1
        GOTO3000
2005    MONTH=1
        IDATE=1
        ITIME=1
3000    WRITE(1802'1)ITIME,IDATE,MONTH,INCTR
        WRITE(1,4)ITIME,IDATE,MONTH,INCTR
        CALL INTEX
        END
```

```
// DUP
*STORECI  I            CLNDR CLNDR         2410
*FILES(1802,HEADR,0)
*CCEND
```

(11) THE CORE LOAD FOR AUTOMATIC DATA TRANSFER ON TAPE

```
// JOB
// FOR AUTO1
*IOCS(TYPEWRITER,KEYBOARD,MAGNETIC TAPE,DISK)
       DEFINE FILE 1801(20,300,U,IFAST)
       DEFINE FILE 1800(20,300,U,ISLOW)
       DEFINE FILE 1802(1,4,U,IHEAD)
       DATA IDOLR/'$$'/
       DIMENSION ITEMP(300)
       COMMON/INSKEL/ISLOW,IFAST
100    FORMAT('DATA TRANSFERED ON TAPE'/'TIME DATE MONTH'
     1  /3I5)
       JSLOW=1
       JFAST=2
       BACK SPACE 14
       ISLOW=1
       IFAST=1
       READ(1802'1)ITIME,IDATE,MONTH, INCTR
```

```
          WRITE(14)ITIME,IDATE,MONTH,JSLOW
          DO1 I=1,20
          READ(1800'ISLOW)ITEMP
          WRITE(14) ITEMP
1         CONTINUE
          WRITE(14)ITIME,IDATE,MONTH,JFAST
          DO2 I=1,20
          READ(1801'IFAST)ITEMP
          WRITE(14) ITEMP
2         CONTINUE
          WRITE(14)IDOLR,ITIME,IDATE,MONTH
          WRITE(1,100)ITIME,IDATE,MONTH
          CALL VIAQ
          END
// DUP
*STORECI  M              AUTO1 AUTO1 AUTO1
*FILES(1800,SFADE;0)
*FILES(1801,FFADE;0)
*FILES(1802,HEADR;0)
*CCEND
```

(12) THE CORE LOAD FOR MANUAL DATA TRANSFER ON TAPE

```
// JOB
// FOR MATRN
*IOCS(TYPEWRITER,KEYBOARD,MAGNETIC TAPE,DISK)
      DEFINEFILE1801(20,300,U,IFAST)
      DEFINEFILE1800(20,300,U,ISLOW)
      DEFINE FILE 1802(1,4,U,IHEAD)
      DATA IDOLR/'$$'/
      DIMENSION ITEMP(300)
      COMMON/INSKEL/ISLOW,IFAST
 99   FORMAT('MOUNT THE TAPE FOR SCATTER COMM. AND MAKE
     1 IT READY')
 100  FORMAT('REMOVE THE TAPE AND SAVE IT')
      JSLOW=1
      JFAST=2
 700  WRITE(1,99)
      PAUSE
 701  CALL          POSTN
      ISLOW=1
      IFAST=1
      READ(1802 1)ITIME,IDATE,MONTH,INCTR
      WRITE(14)ITIME,IDATE,MONTH,JSLOW
      DO 1 I=1,20
      READ(1800'ISLOW)ITEMP
      WRITE(14) ITEMP
```

```
1         CONTINUE
          WRITE(14) ITIME,IDATE,MONTH,JFAST
          DO 2 I=1,20
          READ(1801'IFAST) ITEMP
          WRITE (14)ITEMP
2         CONTINUE
          WRITE(14)IDOLR,ITIME,IDATE,MONTH
          WRITE(1,100)
          REWIND 14
          CALL VIAQ
          END
// FOR POSTN
          SUBROUTINE POSTN
          DATA IDOLR/'$$'/
          DIMENSION IBUFR(4)
          REWIND 14
9         READ(14)IBUFR
          IF(IBUFR(1)-IDOLR)10,12,10
10        DO 11 I=1,20
          READ (14)
11        CONTINUE
          GO TO 9
12        BACKSPACE 14
          RETURN
          END
```

```
*STORECI  M          MATRN MATRN MATRN
*FILES(1800,SFADE,0)
*FILES(1801,FFADE,0)
*FILES(1802,HEADR,0)
*CCEND
// DUP
*SEQCH MATRF MATRN,SCATR
```

(13) SUBROUTINE TO READ SLOW FADE DATA

```
// JOB
// ASM
                    -      ENT         SLOW
                    TEMP   BSS    E    2
                    DATA   EQU         /3FFF
                    AREA1  BSS         1
                    POINT  DC          /0000
                    SLOW   DC          0
                           STD         TEMP
                           STX    L1   XR1+1
                           STX    L2   XR2+1
                           STX    L3   XR3+1
                           LDX    L1   6000
                    START  NOP
                           LDX    L2   3560
                           LIBF        AIPN
                           DC          /1100
```

```
                   DC           AREA1
                   DC           POINT
                   DC           /0000
             LOOPI LIBF         AIPN
                   DC           /0100
                   MDX          LOOPI
                   LD   L       AREA1
                   STO  L1      DATA-6000
             LOOP9 MDX   2      -1
                   MDX          LOOP9
                   MDX   1      -1
                   MDX          START
             XR1   LDX  L1      *-*
             XR2   LDX  L2      *-*
             XR3   LDX  L3      *-*
                   LDD  L       TEMP
                   BSC  I       SLOW
                   END

// DUF
*DELETE      SLOW
*STORE       SLOW
```

(14) SUBROUTINE TO TRANSFER THE SLOW FADE DATA ON TO DISK

```
// FOR
        SUBROUTINE TRAN1
        COMMON/INSKEL/ISLOW,IFAST
        COMMON IDATA (6000)
        ISLOW=1
        IN=1
        IF=300
        DO1 I=1,20
        WRITE(1800' ISLOW)(IDATA(J),J=IN,IF)
        IN=IN+300
        IF=IF+300
1       CONTINUE
        RETURN
        END
// DUP
*DELETE              TRAN1
*STORE               TRAN1
```

(15) SUBROUTINE TO TRANSFER FAST FADE DATA ON TO DISK

```
// FOR
      SUBROUTINE TRAN2
      COMMON/INSKEL/ISLOW,IFAST
      COMMON IDATA (6000)
      IFAST=1
      IN=1
      IF=300
      DO 1I=1,20
      WRITE(1801'IFAST)IDATA(J),J=IN,IF)
      IN=IN+300
      IF=IF+300
1     CONTINUE
      RETURN
      END
// DUP
*DELETE             TRAN2
*STORE              TRAN2
```

(16) SUBROUTINE TO READ FAST FADE DATA

```
// ASM
                          ENT         FAST
                    TEMP  BSS    E    2
```

```
DATA     EQU          /3FFF
AREA1    BSS          1
POINT    DC           /13F8
FAST     DC           0
         STD          TEMP
         STX    L1    XR1+1
         STX    L2    XR2+1
         STX    L3    XR3+1
         LDX    L1    6000
START    NOP
         LDX    L2    170
         LIBF         AIFN
         DC           /1100
         DC           AREA1
         DC           POINT
         DC           /0000
LOOPI    LIBF         AIFN
         DC           /0100
         MDX          LOOPI
         LD     L     AREA1
         STO    L1    DATA-6000
LOOP4    MDX     2    -1
         MDX          LOOP4
         MDX     1    -1
         MDX          START
```

```
                    XR1     LDX  L1     *-*
                    XR2     LDX  L2     *-*
                    XR3     LDX  L3     *-*
                            LDD  L      TEMP
                            BSC  I      FAST
                            END
// DUP
*DELETE             FAST
*STORE              FAST
```

(17)   CORE LOAD SCATR

```
// JOB
// DUP
*STOREDATED     0      HEADR     001
// DUP
*STOREDATAD     0      SFADE     020
// DUP
*STOREDATAD     0      FFADE     020
// JOB
// FOR DIGIT
*IOCS(DISK)
       DEFINE FILE 1800(20,300,U,ISLOW)
       DEFINE FILE 1801(20,300,U,IFAST)
       DEFINE FILE 1802(1,4,U,IHEAD)
```

```
          EXTERNAL AUTO1
          EXTERNAL MATRF
          COMMON/INSKEL/ISLOW,IFAST
          COMMON IDATA(6000)
          CALL SLOW
          CALL TRAN1
          CALL FAST
          CALL TRAN2
          IHEAD=1
          READ(1802'IHEAD)ITIME,IDATE,MONTH,INCTR
          GOTO(100,101),INCTR
100       CALL QUEUE(MATRF,1,0)
          GOTO 300
101       CALL QUEUE(AUTO1,1,0)
300       CALL ENDTS
          CALL ENTEX
          END
// DUP
*DELETE                 DIGIT
*STORE                  DIGIT
// JOB
// DUP
*STORECI  I             SCATR DIGIT SCATR 2405
*LOCALSLOW,TRAN1,FAST,TRAN2
```

```
*FILES(1800,SFADE,0)

*FILES(1801,FFADE,0)

*FILES(1802,HEADR,0)

*CCEND
```

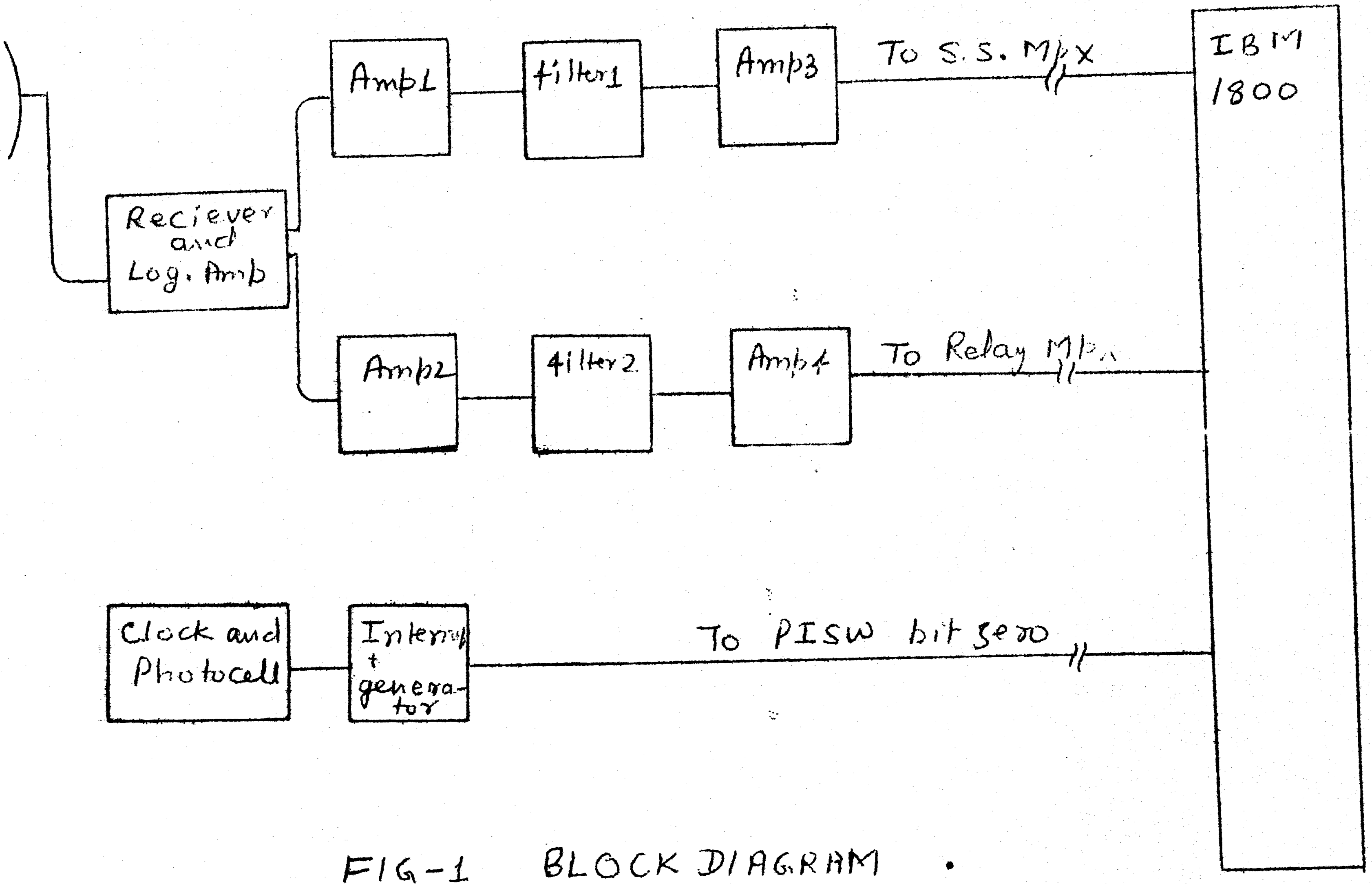


FIG-1 BLOCK DIAGRAM.

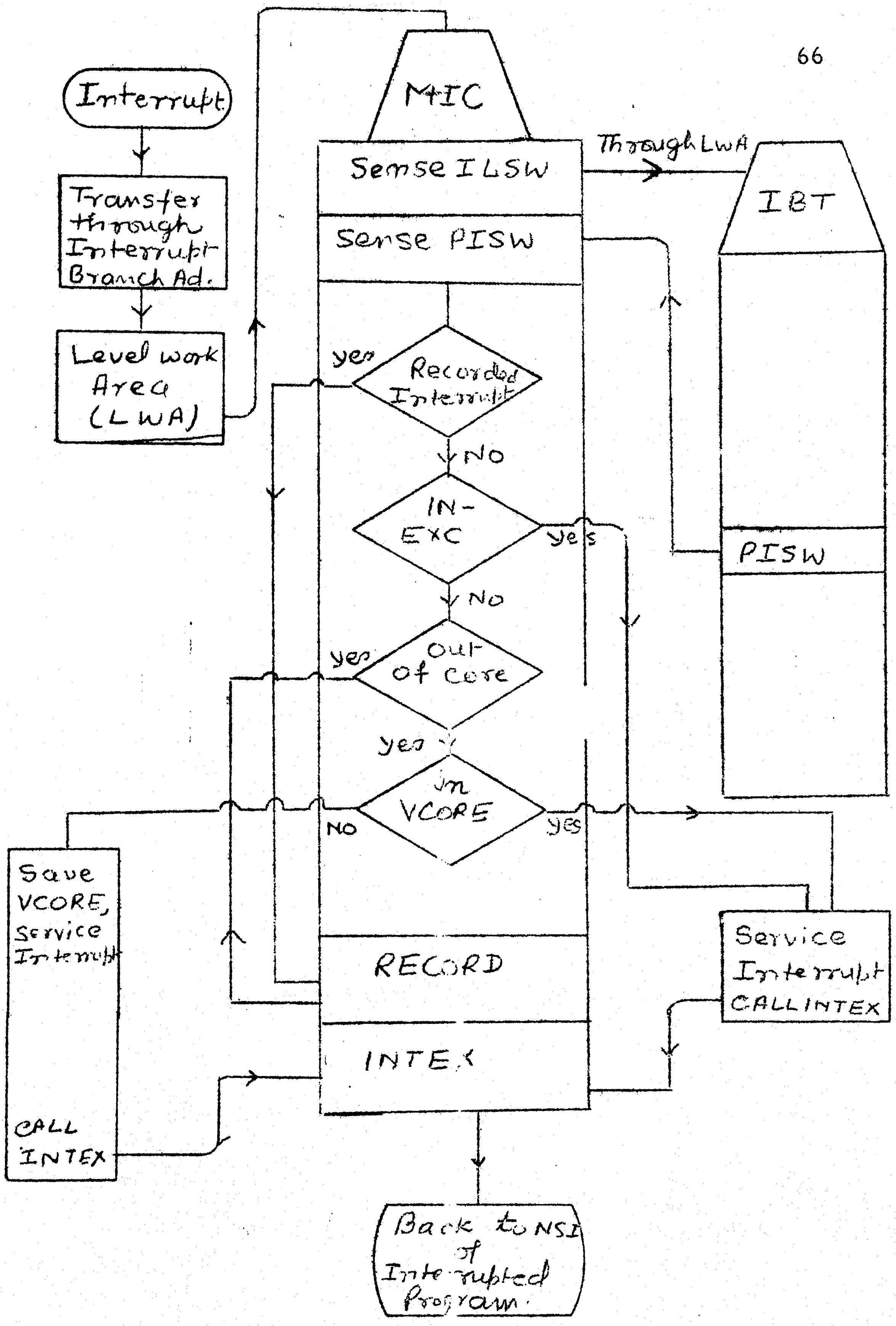


FIG. 2. SERVICING OF INTERRUPTS

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
14 bit format	±	DATA word														I
11 bit format	±	DATA word											x	x	x	I
8 bit format	±	DATA word								x	x	x	x	x	x	I

FIG-3 ADC DATA WORD

x ≡ not used

I ≡ Overflow

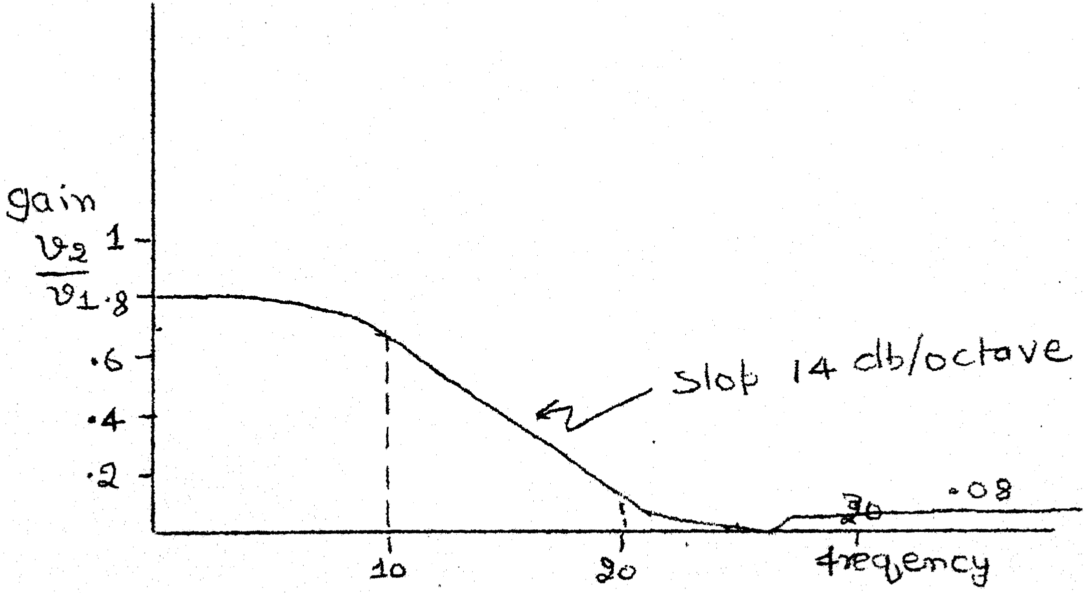


FIG 5(a) FILTER CHARACTERSTICS.

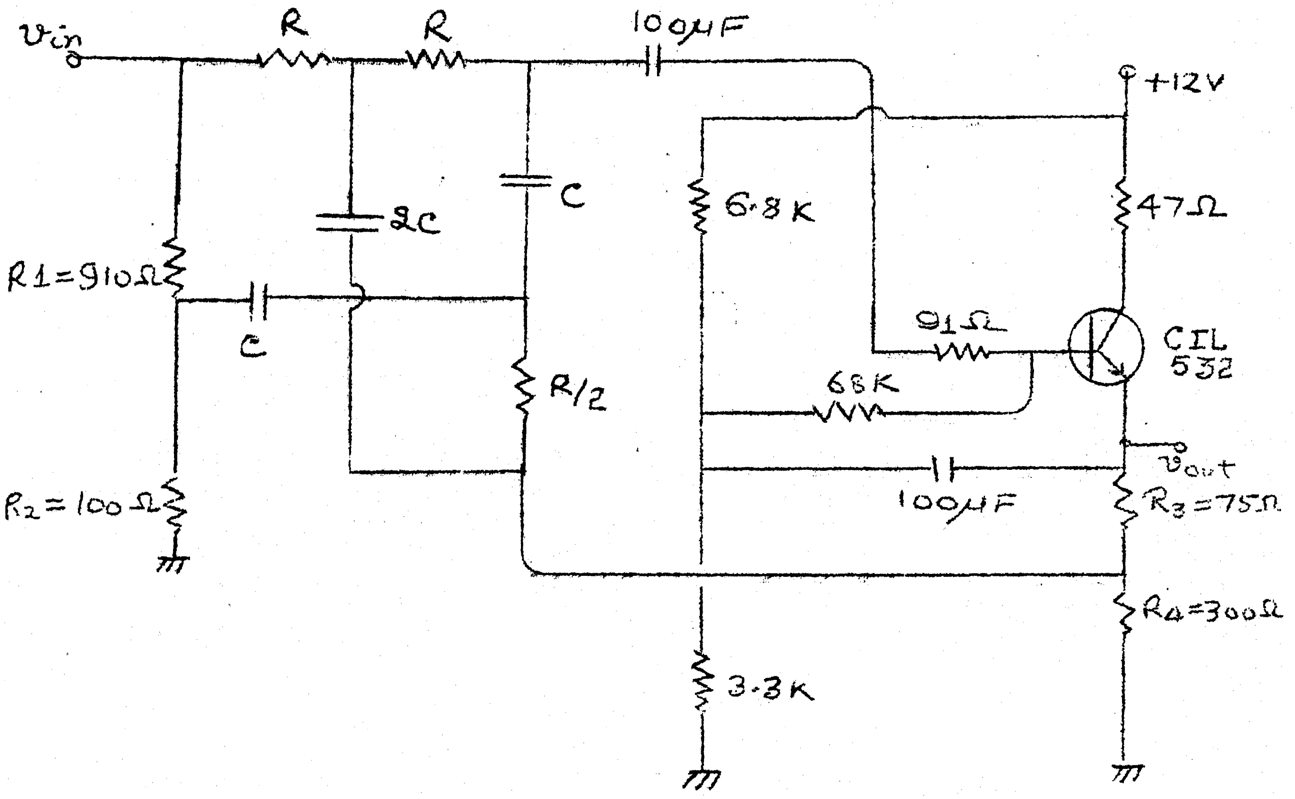
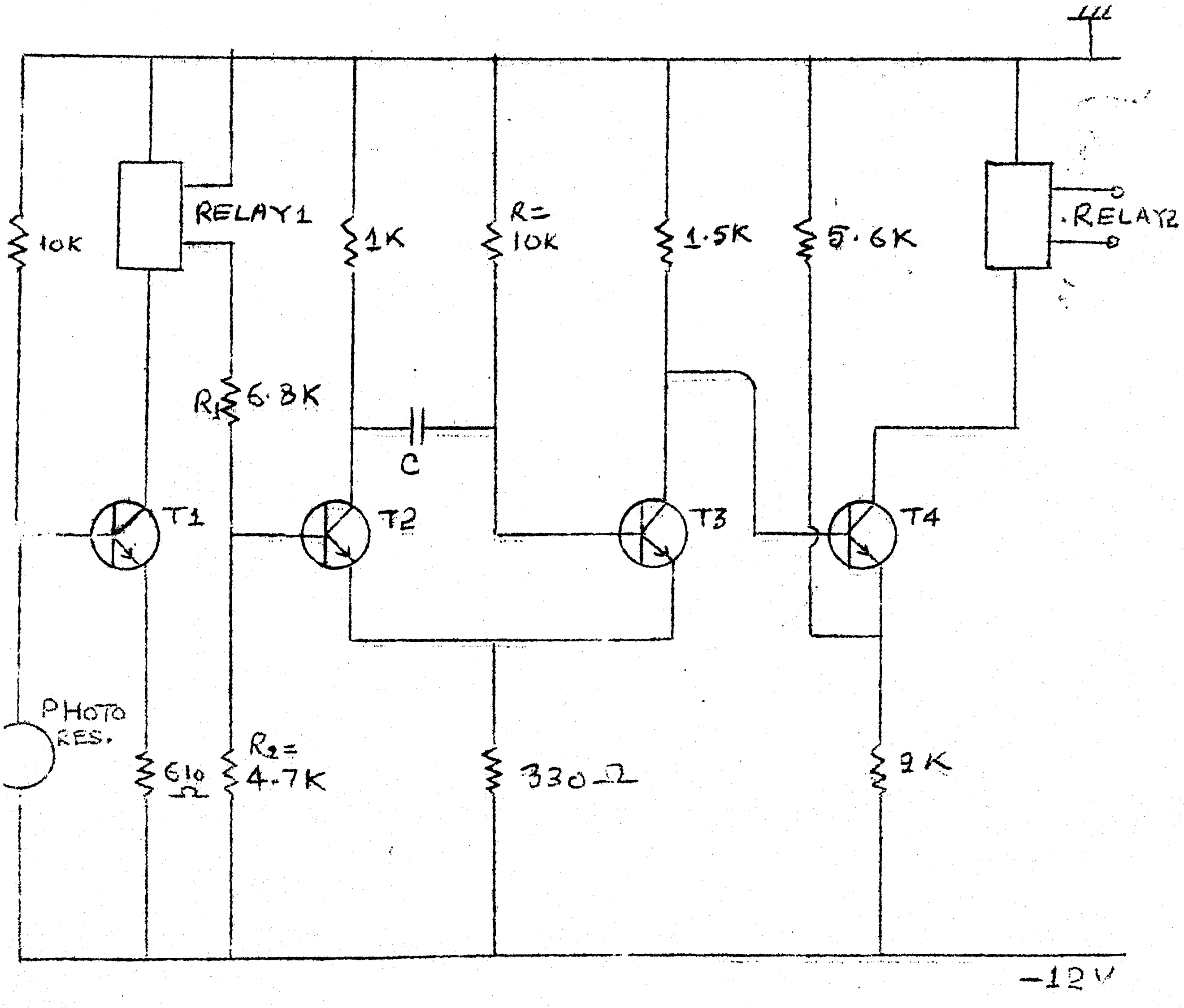


FIG.5(a) – LOW PASS FILTER



.7RC = 4 m sec.

C = .6 μF

FIG-6 INTERRUPT GENERATOR

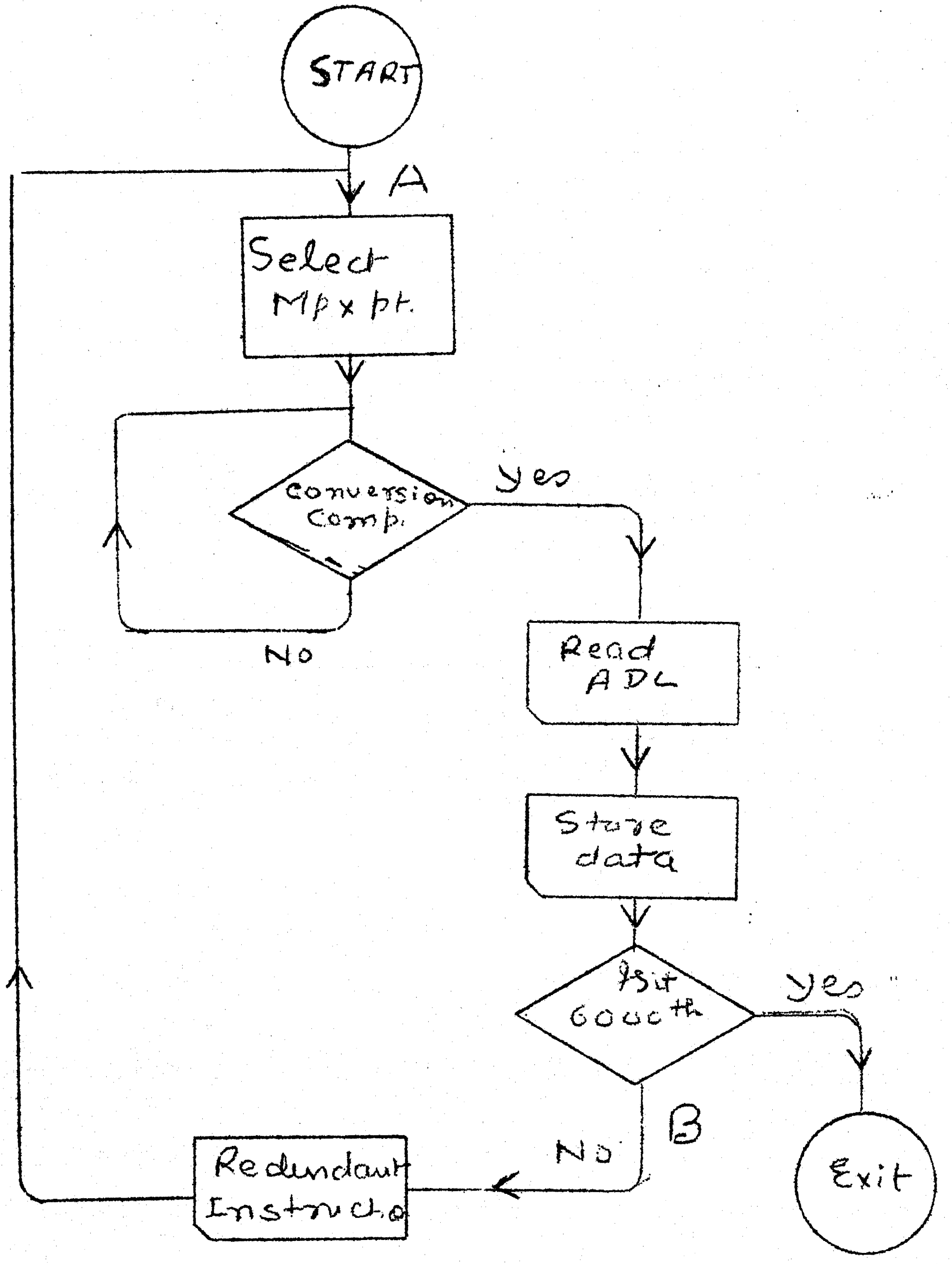


FIG-7 - READING ANALOG INPUT

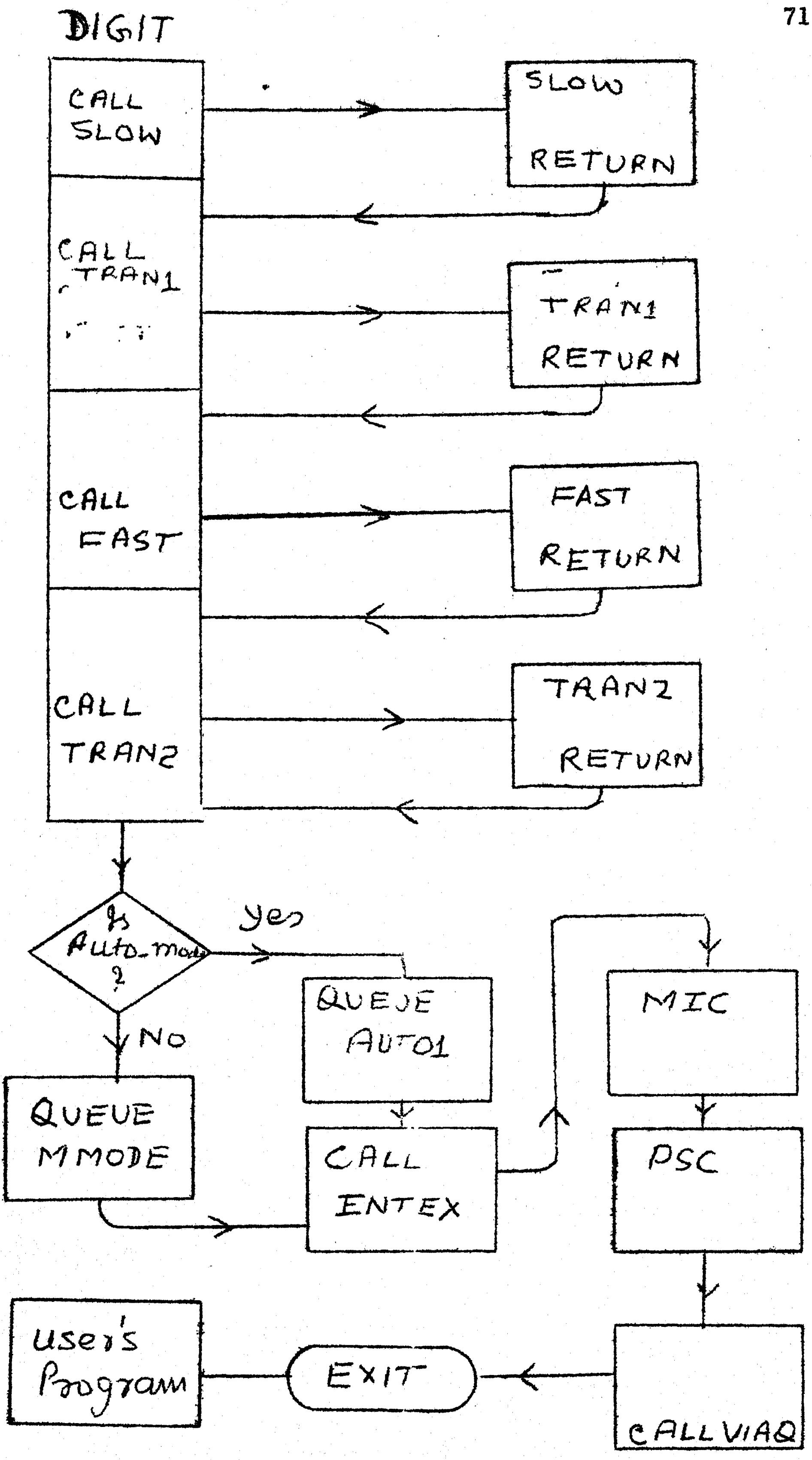


FIG-8 FLOW OF CONTROL IN 'SCATR'

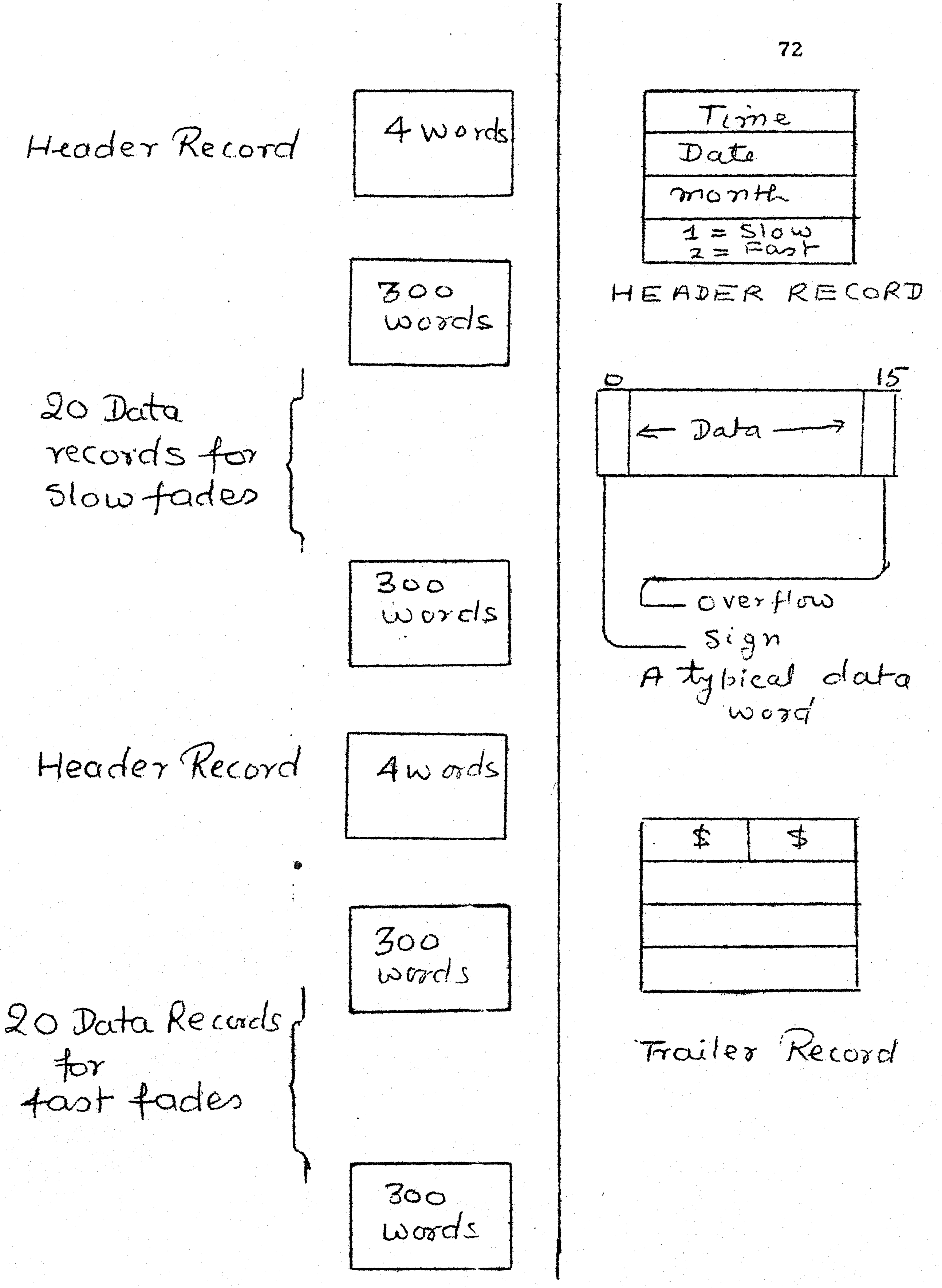


FIG-9 - MAGNETIC TAPE FILE (Per hour)

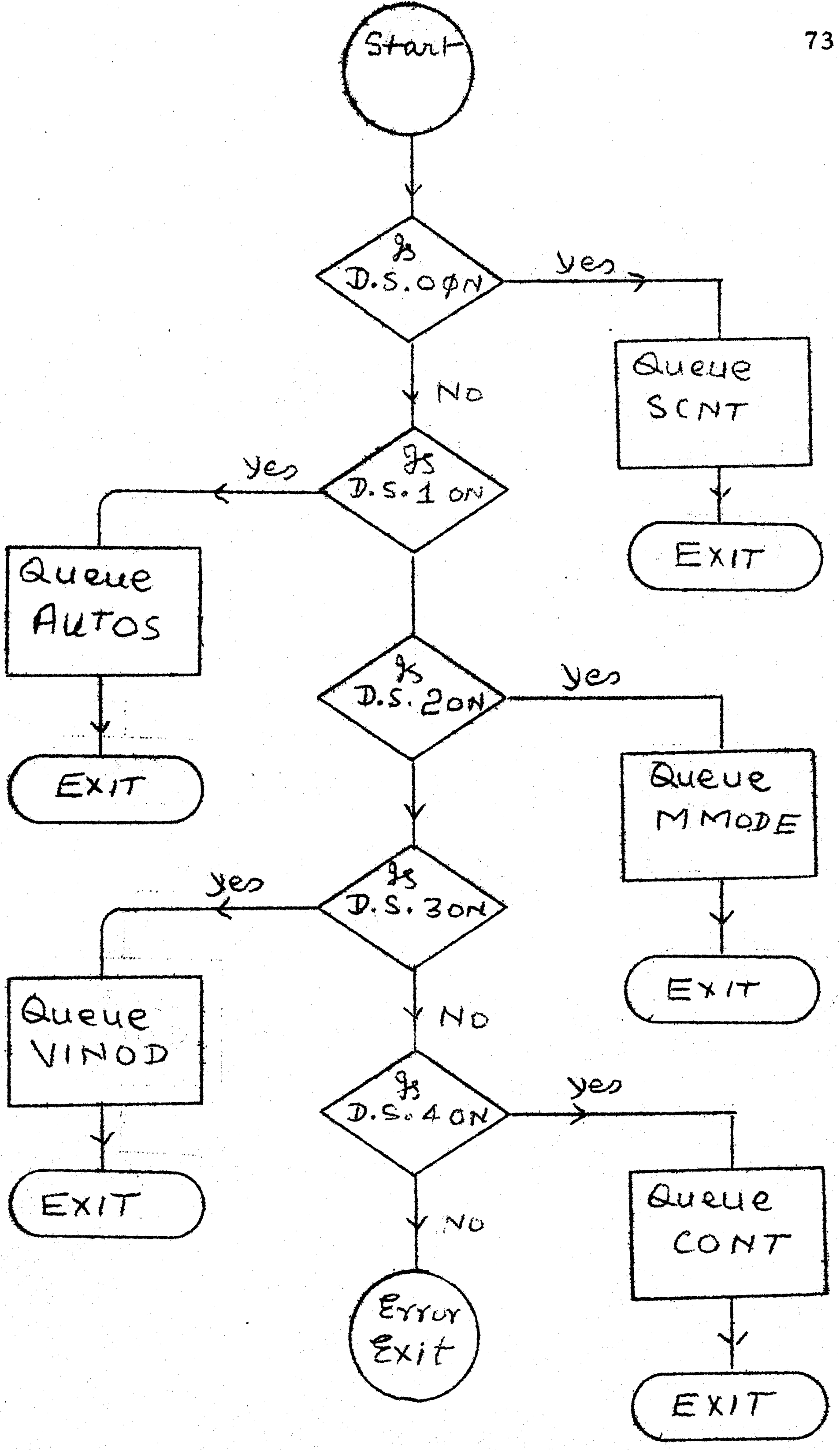


FIG-10 CORE LOAD INTO7

TABLE 1

S No.	Voltage	Change in voltage in one sampling time
1	-.114	.117
2	.003	.182
3	.185	.187
4	.372	.189
5	.561	.188
6	.749	.190
7	.939	.178
8	.117	.192
9	1.309	.185
10	1.486	.177
11	1.671	.185
12	1.855	.190
13	2.045	.180
14	2.225	.181
15	2.406	.179
16	2.585	.187
17	2.772	.188
18	2.960	.179
19	3.139	.120
20	3.019	.374
21	2.645	.403
22	2.242	.402
23	1.840	.404
24	1.438	.418
25	1.020	.409
26	0.611	.421
27	0.190	.303
28	- 0.113	

REFERENCES

1 M.Tech Thesis Jayaram Murthy & Rastogi I.I.T./K-1969.

2. IBM 1800 Functional Characteristics Manual

3. IBM 1800 Physical Planning Manual

4. IBM 1800 Time Sharing Executive, Concepts and Techniques

5. IBM 1800 Time Sharing Executive, Subroutine Library.

6. IBM 1800 Time Sharing Executive, Operating Procedures

7. IBM 1800 Time Sharing Executive, Program Logic Manual

8. Pulse Techniques, Millman and Taub

9. Farrer. W; Electronic Engineering,V 33,No.-4 1967, p 219.